



# Planning Modelling Language

July 2010



---

<b>1</b>	<b>Planning Modelling Language .....</b>	<b>3</b>
1.1	Timeline Entry .....	3
1.2	Structure .....	4
1.2.1	Task .....	4
1.2.2	Group .....	4
1.3	Resources .....	5
1.4	Constraints .....	6
1.4.1	Assignment Restriction .....	6
1.4.2	Demand .....	6
1.4.3	Group Scheduling Restriction .....	7
1.4.4	Time Dependency .....	7
1.4.5	Resource Comparison .....	8
1.4.6	Resource Modification .....	9
1.4.7	Resource Bound .....	11
1.4.8	Suitability .....	11

# 1 Planning Modelling Language

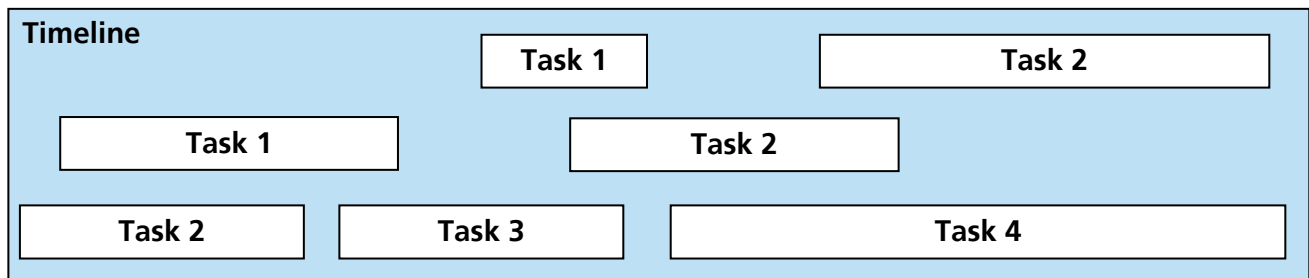
The goal of the scheduling process is to generate a timeline, i.e. a set of timeline entries, where each timeline entry refers to a certain activity. This timeline must be conflict free in order to be executable. To ensure this, resources are defined, which describe the state of the system over time and constraints on the tasks' timeline entries may be defined, which shall reflect the real world's constraints.

## 1.1 Timeline Entry

A timeline entry specifies that a task (see [Task](#)) is executed during a certain time interval. Its main properties are:

- Start Time
- Duration
- Task

All timeline entries form the timeline:



## 1.2 Structure

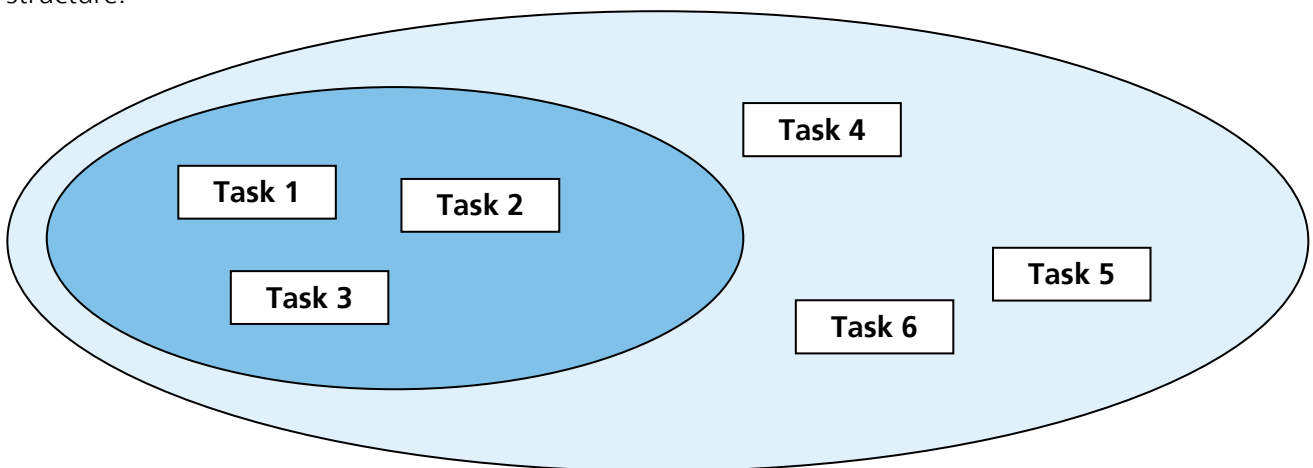
### 1.2.1 Task

A task represents a certain activity and may be mapped on the timeline by creating a timeline entry. Its main properties are:

- Name (indicating its real world counterpart)
- Priority

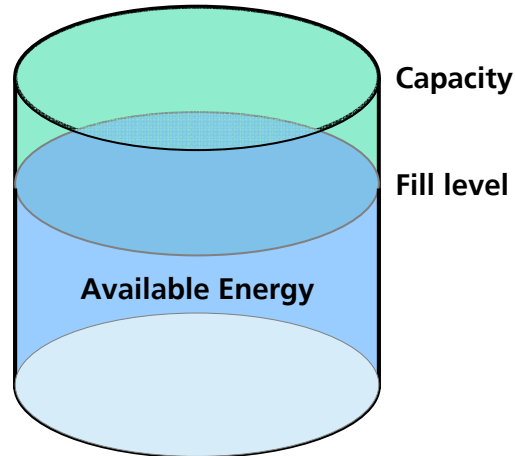
### 1.2.2 Group

In order to enable the algorithm to identify tasks, which belong together, multiple tasks may be put together in a group. A group may also contain other groups, which allows building up a hierarchical structure:



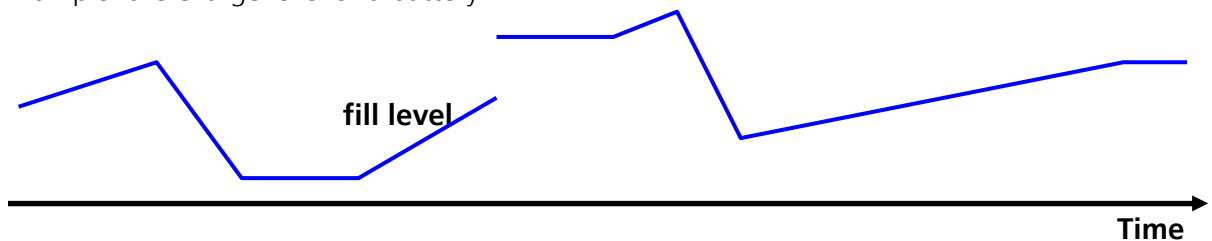
### 1.3 Resources

A resource represents a certain scheduling relevant aspect of the system, e.g. the charge level of a battery or the orientation of a satellite:

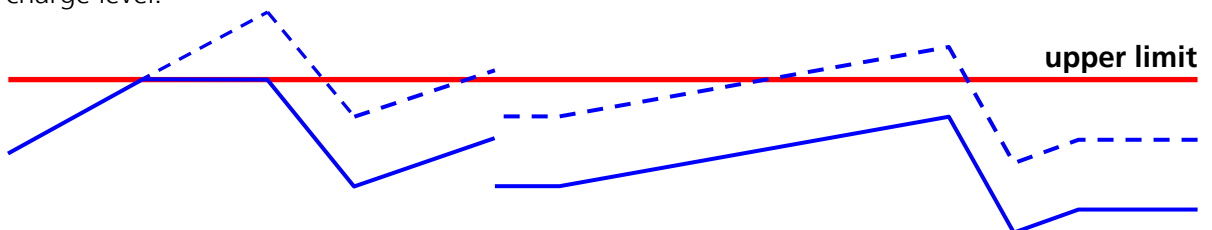


A resource's main features are three time profiles, which describe the state of the resource subject to time:

- **Fill level**  
the fill level describes the measurable value of the resource at a given time.  
Example: the charge level of a battery



- **Upper limit [optional]**  
the upper limit represents the maximum value, the fill level may reach. In case the fill level would reach a higher value at a time T, its value is automatically reduced to the upper limit's value at time T  
Example: a battery's capacity supplies an upper limit for the charge level of the battery. The dashed line represents the charge level in case the battery's capacity would be infinite. Given a capacity according to the upper limit, the drawn through line represents the actually achieved charge level.



- **Lower limit [optional]**  
similar to the upper limit, but supplies a minimum value for fill level instead of a maximum value

## 1.4 Constraints

A constraint describes a rule, which the scheduling engine shall obey while generating the timeline.

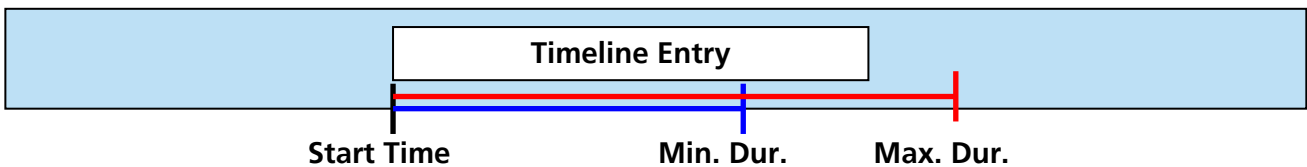
### 1.4.1 Assignment Restriction

An assignment restriction has the following properties:

- Task
- Minimum Duration
- Maximum Duration

Interpretation:

Each timeline entry of Task must have a duration in [Minimum Duration , Maximum Duration]



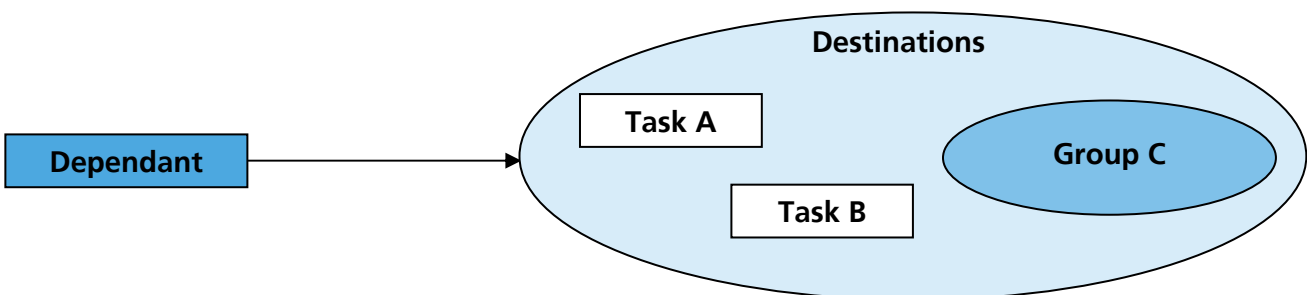
### 1.4.2 Demand

A demand has the following properties:

- A task, called Dependant
- A set of tasks and groups, called Destinations

Interpretation:

The dependant must not be scheduled unless at least one of the destinations is scheduled



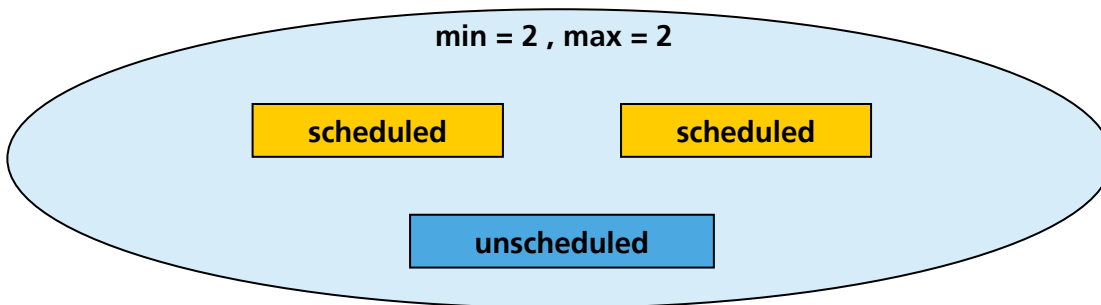
### 1.4.3 Group Scheduling Restriction

A group scheduling restriction has the following properties:

- Group
- Minimum Number To Schedule [min]
- Maximum Number To Schedule [max]

Interpretation:

The number  $N$  of scheduled elements on Group must satisfy  $(N = 0)$  or  $(\min \leq N \leq \max)$



### 1.4.4 Time Dependency

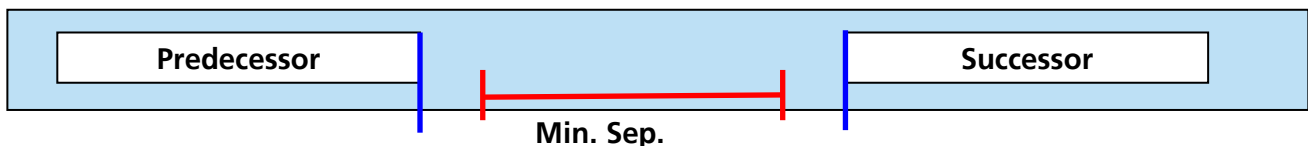
A time dependency has the following properties:

- A task called Predecessor
- A task called Successor
- A duration, called Minimum Separation
- A value from {start , end}, called Predecessor Reference Type [ref\_p]
- A value from {start , end}, called Successor Reference Type [ref\_s]

Interpretation:

For all pairs  $(TE_p, TE_s)$  with  $TE_p$  is a timeline entry of Predecessor and  $TE_s$  is a timeline entry of Successor:

1. Depending on ref\_p, let  $T_p$  denote the start time resp. the end time of  $TE_p$
2. Depending on ref\_s, let  $T_s$  denote the start time resp. the end time of  $TE_s$
3. Then the following must be satisfied:  $T_p \leq T_s$



**Remark:** A maximum separation may be defined by swapping the roles of predecessor and successor and negating the minimum separation.

### 1.4.5 Resource Comparison

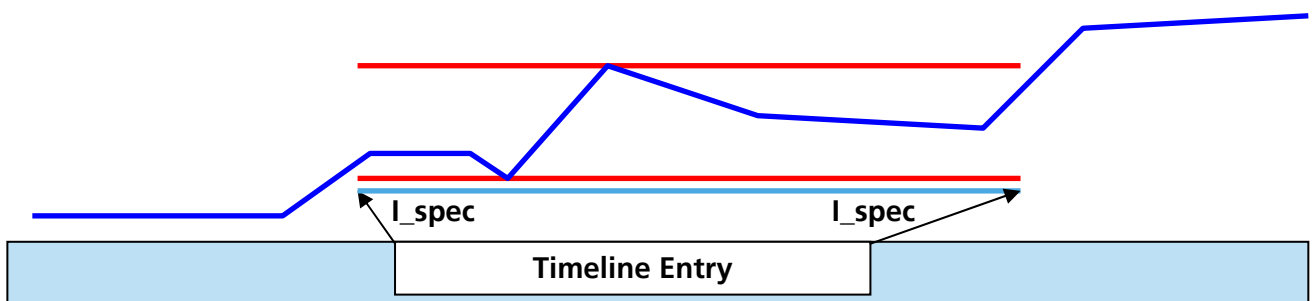
A resource comparison has the following properties:

- Task
- Resource
- Interval Specifier [I\_spec], consisting of
  1. Start time reference (start or end)
  2. Start time offset (a duration)
  3. End time reference (start or end)
  4. End time offset (a duration)
  5. Include End (a bool)
- A duration profile, called Upper Bound, which must be defined on the interval [0 , +infinity]
- A duration profile, called Lower Bound, which must be defined on the interval [0 , +infinity]

Interpretation:

For each timeline entry TE of Task:

- I\_spec converts TE to a time interval I (The start time of I is given by adding an offset to either the start time or the end time of TE. The end time is specified similarly.)
- I.duration shall denote the size of the interval I and I.Start shall denote the start time of I
- The Upper Bound is converted to the effective upper bound UpperEff by identifying the duration 0 with I.Start and restricting the resulting profile to the interval I.
- The Lower Bound is converted to the effective lower bound LowerEff by identifying the duration 0 with I.Start and restricting the resulting profile to the interval I.
- Denote the Resource’s fill level with FL. Then the following must hold:  
 $LowerEff(T) \leq FL(T) \leq UpperEff(T)$  for all T in I





### 1.4.6 Resource Modification

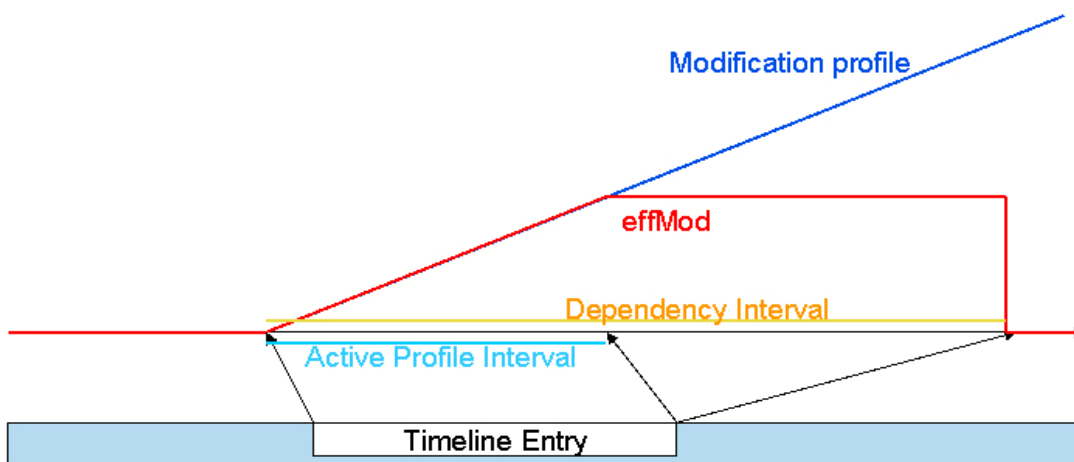
A resource modification has the following properties:

- Task
- Resource
- Start Specifier [Min\_spec], consisting of
  - Start time reference (start or end)
  - Start time offset (a duration)
- Active Profile End Specifier [Prof\_Max\_spec], consisting of
  - End time reference (start or end)
  - End time offset (a duration)
  - IncludeEnd (a bool)
- Dependency End Specifier [Dep\_Max\_spec], consisting of
  - End time reference (start or end, must be 'end' in case Prof\_Max\_spec.EndTimeReference = end)
  - End time offset (a duration, must be greater or equal Prof\_Max\_spec.EndTimeOffset)
  - IncludeEnd (a bool)

Min\_spec and Prof\_Max\_spec define the Active Profile Interval Specification, called A\_spec.  
 Min\_spec and Dep\_Max\_spec define the Dependency Interval Specification, called D\_spec.

Interpretation:

Let TE denote a newly created timeline entry of Task. Similar to chapter [Resource Comparison](#), the TimelineEntry and A\_spec convert the Modification Profile to a time profile, called effective modification (effMod). In case the interval ModInterval given by D\_spec and TimelineEntry is greater than A\_spec, effMod is extended to ModInterval using its endValue:



This time profile must be added to the Resource's fill level profile.

Two examples follow:

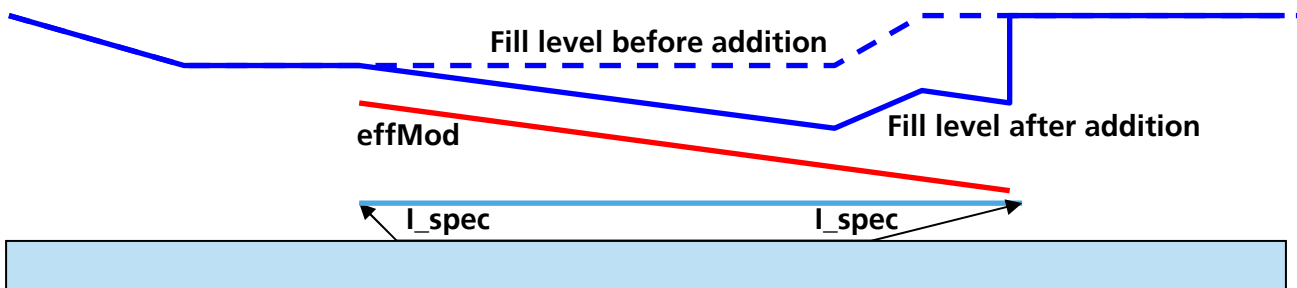
**1.4.6.1 Resource Allocation (Resource Modification with Dependency Interval = Active Profile Interval)**

A resource allocation has the following properties:

- Task
- Resource
- Interval Specifier [I\_sepc]
- A duration profile, called Modification Profile [prof], which must be defined on the interval [0 , +infinity]

Interpretation:

Let TE denote a newly created timeline entry of Task. Similar to chapter [Resource Comparison](#), the TimelineEntry and I\_sepc convert the Modification Profile to a time profile, called effective modification (effMod). This time profile must be added to the Resource's fill level profile.



**Remark:**

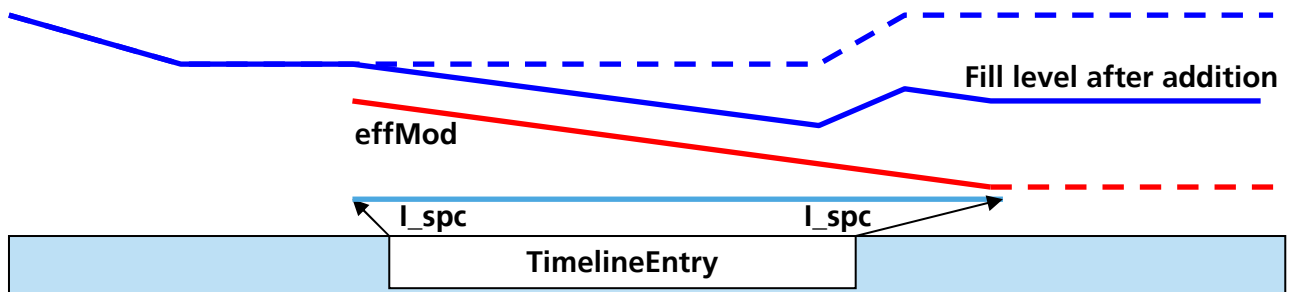
This type of constraint can cause a conflict only in case there exist bounds on the Resource (see [Resource Bound](#) and [Resource Comparison](#))

**1.4.6.2 Resource Accumulation (Resource Modification with Dependency Interval End = +infinity)**

A resource accumulation is similar to a resource allocation with one difference:

The effective modification is not restricted to the interval I, but to the interval [I.start , +infinity].

During the interval [I.end , +infinity], effMod has value effMod(I.end):



**Remark:**

This type of constraint can cause a conflict only in case there exist bounds on the Resource (see [Resource Bound](#) and [Resource Comparison](#))

### 1.4.7 Resource Bound

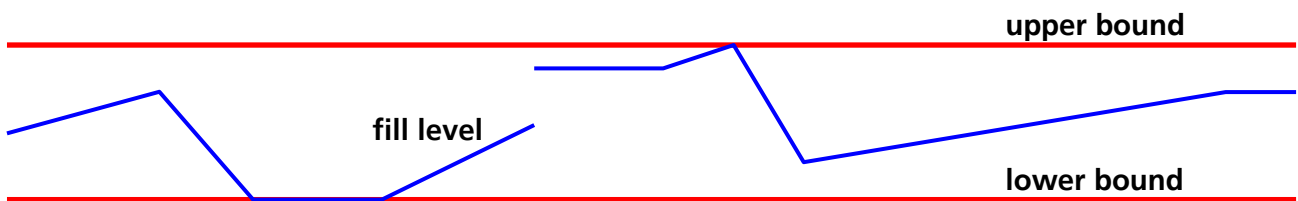
A resource bound has the following properties:

- Resource
- A time profile, called Upper Bound
- A time profile, called Lower Bound

Interpretation:

The fill level profile of the Resource must satisfy

$$\text{Lower Bound}(T) \leq \text{fill level}(T) \leq \text{Upper Bound}(T)$$



**Remark:**

In contrast to resource limits (see [Resources](#)), where the calculation logic assures that the fill level stays in between the specified limits, the fill level may exceed the bounds. This however would represent a conflict and therefore must be avoided by the scheduling engine.

### 1.4.8 Suitability

A suitability has the following properties:

- Task
- Resource
- Interval Specifier [I\_spec]

Interpretation:

I\_spec converts a timeline entry of Task to a time interval I (similar as in [Resource Comparison](#)). The timeline entry shall be selected such that the fill level of Resource, restricted to I, is maximized. What norm to use for this optimization, may be selected from:

- minimum value
- maximum value
- integral

